

# DNA Steganalysis Using Deep Recurrent Neural Networks

Ho Bae

Interdisciplinary Program in Bioinformatics  
Seoul National University  
Seoul 08826, South Korea

Sunyoung Kwon

Department of Electrical and Computer Engineering  
Seoul National University  
Seoul 08826, South Korea

Byunghan Lee

Department of Electrical and Computer Engineering  
Seoul National University  
Seoul 08826, South Korea

Sungroh Yoon\*

Department of Electrical and Computer Engineering  
& Interdisciplinary Program in Bioinformatics  
Seoul National University  
Seoul 08826, South Korea  
e-mail: sryoon@snu.ac.kr

## ABSTRACT

The technique of hiding messages in digital data is called a steganography technique. With improved sequencing techniques, increasing attempts have been conducted to hide hidden messages in deoxyribonucleic acid (DNA) sequences which have been become a medium for steganography. Many detection schemes have developed for conventional digital data, but these schemes not applicable to DNA sequences because of DNA's complex internal structures. In this paper, we propose the first DNA steganalysis framework for detecting hidden messages and conduct an experiment based on the random oracle model. Among the suitable models for the framework, splice junction classification using deep recurrent neural networks (RNNs) is most appropriate for performing DNA steganalysis. In our DNA steganography approach, we extract the hidden layer composed of RNNs to model the internal structure of a DNA sequence. We provide security for steganography schemes based on mutual entropy and provide simulation results that illustrate how our model detects hidden messages, independent of regions of a targeted reference genome. We apply our method to human genome datasets and determine that hidden messages in DNA sequences with a minimum sample size of 100 are detectable, regardless of the presence of hidden regions.

## KEYWORDS

Deep recurrent neural network, DNA steganography, DNA steganalysis

## 1 INTRODUCTION

Discovering a new covert medium is one of the fundamental problems in espionage [5]. With the advent of genotype-chip technologies, individual genotyping has become less expensive [15], and deoxyribonucleic acid (DNA) in turn has become covert medium. A gram of DNA contains approximately 1021 DNA bases (108 terabytes), which suggests that only a few grams of DNA can store all information available [18]. DNA and ribonucleic acid (RNA) are attractive for data storage because they allow large amounts of

data to be stockpiled and because they exceed the current storage capacities of resources such as electronic and magnetic media.

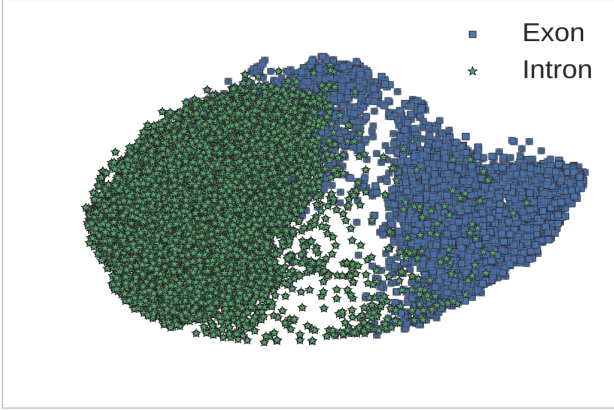
Yachie et al. [55] demonstrated that living organisms can be used as a data storage media by inserting artificial DNA into artificial genomes and using a substitution cipher coding scheme. This technique is reproducible and successfully inserts four watermarks into the cell of a living organism [19]. Several other encoding schemes have been proposed [10, 51]. The DNA-Crypt coding scheme [23] translates a message into 5-bit sequences, and the ASCII coding scheme [26] translates words into their ASCII representation, converts them from decimals to binary, and then replaces 00 with adenine (A), 01 with cytosine (C), 10 with guanine (G), and 11 with thymine (T).

With the development of DNA hiding schemes, hidden messages can be concealed in an organism's DNA using steganography. Any digital data such as text or images can be inserted into a DNA sequence using the aforementioned coding schemes. Thus, the development of forensic tools, namely, steganalysis for DNA steganography, was inevitable. However, the current versions of detection schemes were developed to detect hidden messages in text, images, audio, and video media. For example, Bennett [7] exploits text material according to letter frequency, word frequency, grammar style, semantic continuity, and logical methodologies. The advent of detection techniques based on statistical analysis, neural networks, and genetic algorithm [38] methods have been developed for today's most common covert objects of digital images, video, and audio.

Although detection schemes exist, conventional detection methods have not been applied to DNA sequences. We propose some methods for DNA steganalysis using the sequence analysis framework and denoising tools. However, these methods are often restricted by the robustness of the target prediction, and miss important signals during splitting and merging procedures [33]. To overcome the aforementioned limitations of frequency analysis, sequence alignment, and denoising tools, we propose a method based on introns/exons analysis and our previous work using splice junction prediction (GT or AT) which has gained popularity as an alternative to the machine learning approach.

Our method uses recurrent neural networks (RNNs), which are artificial neural networks in which connections between units form a directed cycle. These feedback loops create an internal network

\*Corresponding author should be addressed. S. Yoon is also with the Department of Neurology and Neurological Sciences, Stanford University, Stanford, CA 94305, USA.



**Figure 1: The learned representations for each coding and non-coding region projected into a two-dimensional (2-D) space using t-SNE [37]. The representation is based on sequence-to-sequence learning using autoencoder and stacked RNNs. The input layer is connected to the first layer of an autoencoder which is unsupervised pre-training of sequence-to-sequence learning. The second layer is RNNs layer to construct the model for determining patterns between introns and exons. The learned representations of introns and exons which are represented in green stars-mark and blue squares-mark, respectively (best viewed in color).**

state that allows RNNs to exhibit dynamic temporal behavior. Learning temporal behavior enables DNA sequence pattern recognition to investigate omics data [31, 34, 41]. Among the various sequence modeling tasks, RNNs [21, 39] are used to obtain a robust prediction result by learning the internal structure (introns and exons) of DNA sequences.

In this paper, we propose a DNA steganalysis method using sequence analysis (i.e., denoising tools [32, 43, 48, 57], and splice junction classification [31, 34]) for detecting hidden messages. We investigate a variety of machine learning methods for splice junction classification for detecting hidden messages and compare them with denoising tools and sequence analysis. Finally, we demonstrate that our proposed method using RNNs achieves the best performance when compared to other learning methods including support vector machine (SVM), random forest, and adaptive boosting for the current DNA steganography schemes listed in Table 1.

The remainder of this paper is organized as follows. In Section 2, we describe the concept of steganography and steganalysis and their association to DNA internal structures. In Section 3, we provide our proposed RNNs model architecture and also show that current DNA steganography schemes are theoretically not secure. In Section 4, we discuss experimental results from various machine learning, denoising, and sequence alignment methods. We conclude our paper in Section 5.

**Table 1: Existing DNA steganography schemes.**

Coding Scheme	Hiding Message Region	Modification Rate
Substitution based on primers of microdots [14]	Introns & Exons	20 %
DNA cryptosystem with one-time pads [35]	Introns & Exons	$\leq 9\%$
Insert to artificial DNA strand [54]	Introns & Exons	20%
Improved DNA cryptosystem with one-time pads [18]	Introns & Exons	-
Insertion based algorithm namely Arita [3]	Exons	$\leq 5\%$
Applied existing encryption algorithms to DNA [23]	Exons	$\leq 8\%$
Substitution and insertion based algorithm [47]	Introns & Exons	-

## 2 BACKGROUND

In molecular biology, DNA information flows from DNA to RNA by means of transcription and to protein by translation. Four types of nucleotides, (A, C, G, and T) are translated into 20 types of amino acids according to the genetic code, except for three stop codons (TAA, TAG, and TGA). An mRNA corresponds to DNA except that in RNA, T is replaced by uracil (U). With 4 nucleotides,  $4^3 = 64$  possible combinations of codon exist, whereas only 20 amino acids exist.

The translation of multiple codons into single amino acid allows for degeneracy in the genetic code. For example threonine can be derived from ACU, ACC, ACA, and ACG and the third base does not exert any influence on the translation to protein sequences. These mutations of synonymous substitution allow potential exploits to embed information in the DNA sequences. A technique that hides secret messages through original files such that the existence of the secret message is unknown is called steganography. We describe this technique in detail in the following section.

### 2.1 Steganography

Original files can be text, images, audio, or DNA sequences. After hidden messages are embedded, an original file is referred to as a steged-medium. It is used as a communication channel for hiding the existence of the message itself, thus making it difficult for a third party to find the message. In addition, the steganography scheme can also be used as watermarking for intellectual property. The signature in watermarking indicates the ownership of the data and ensures copyright protection.

Hiding a message into a DNA sequence has become topic of scientific investigation since it was first proposed by Clelland et al. [14] and has gained popularity over the last 10 years. The scheme for hiding data in a DNA sequence was first proposed using a microdot technique[14]. Leier et al. [35] proposed a robust encryption scheme using a primer as the key sequence. A public DNA sequence is used as a reference and the selected primer and encrypted sequences are sent to the receiver. The primer is short complementary DNA sequences that depend on each proposed scheme. Leier et al. [35] uses complementary pairs of (A  $\rightarrow$  T, T  $\rightarrow$  A, C  $\rightarrow$  G, and G  $\rightarrow$  C) whereas that Shiu et al. [47] uses complementary pairs of (A  $\rightarrow$  C, C  $\rightarrow$  G, G  $\rightarrow$  T, and T  $\rightarrow$  A) for an encryption scheme. These DNA steganography schemes are claimed to be secure if primers and a reference sequence are not known. However, we will show that hidden messages are detectable without knowing primers despite their claim (see section 3).

## 2.2 Steganalysis

Steganalysis operates in two modes. The first method involves removing the watermark(s) signal, which is processed by scanning. The second method involves compressing while detecting a portion of a ciphertext by frequency or entropy measurements. Traditional data hiding approaches usually embed a secret message into the images and audio files [56, 58]. However, with the advent of detection schemes, even very small distortions of images can easily be detected [53]. However, the conventional DNA steganalysis methods do not work with DNA steganography. This is because a biochemical approach is required to extract DNA sequences and a biological process (transcription/translation) makes it difficult to encrypt/decrypt secret messages.

## 2.3 Determination of Message-Hiding Regions

Genomic sequences contain both exons (coding regions) and introns (non-coding regions). These two regions are utilized depending on whether the task is one of data storage or transport. Intron regions are a candidate for transportation because not all regions are transcribed to RNA regions and removed by splicing [28, 36] during transcription. This functionality provides vast space for hiding data, creating potential covert channels. By contrast, the data must be resistant to degradation or truncation in data storage (watermarking). Exons are one suitable candidate for storage because the sequences are carried over after the translation and transcription processes [46]. These two components of internal structures in eukaryote genes are involved in DNA steganography as payload (watermarking) or carrier (covert channel). The use of DNA sequences as payloads or carriers varies from algorithm to algorithm, as shown in Table 1.

Figure 1 shows the learned representations for each coding and non-coding region projected into a two-dimensional (2-D) space using t-SNE [37]. The learned representations of introns and exons are denoted by green stars and blue squares, respectively. Some parts of the regions overlap in a 2-D space which can be interpreted to mean that shared patterns exist between introns and exons. The position of introns and exons will change if hidden messages are embedded. Hidden messages are readily detectable if a clear separation exists, but the shared patterns make detection difficult. A distinct, clear separation somehow overcomes the limited representation of DNA sequences [18]. Thus, clear separations of these shared patterns are one important factor in constructing a classification model for detecting hidden messages.

## 3 METHODS

Our proposed method uses RNNs to detect hidden messages in DNA and is a new method in the field of steganalysis. Figure 2 shows the overview of our proposed method, which consists of three phases: 1) species identification, 2) training, and 3) detection. In the species identification, for a given query  $m$ , we discover its candidate species regardless of whether the query has hidden messages. In the model training phase, we build a model using its reference sequences  $D_i \in \mathbf{D}$  for the chosen species  $i$  from the public database. In the detection phase, we obtain a probability of random sequence  $m_i \in D_i$  from the training phase and repeat until it obtains an

overwhelming probability. Finally, we obtain the test probability with the given query  $m$  during the testing phase.

## 3.1 Notations

We use the standard terminology of information hiding [2] to provide a brief explanation of this section. Two hypothetical parties, Alice and Bob, wish to communicate between two restricted sequencing laboratories. They are allowed to communicate only through a third party. A message between them is not sent if the third party finds their secret message in a DNA sequence. Alice and Bob use existing steganography schemes to hide their secret message, and the third party uses our proposed model to detect the secret message. Details of our proposed method are described in the following section. The notations used in this paper are as follows:

- $\mathbf{D} = \{D_1, \dots, D_n\}$  is a set of DNA sequences of  $n$  species in which  $D_i$  is the  $i$ -th species of DNA sequences representation.
- $\hat{\mathbf{D}} = \{\hat{D}_1, \dots, \hat{D}_n\}$  is a set of DNA sequences of  $n$  species. Hidden messages are embedded for some species  $\hat{D}_i$  where  $\hat{D}_i$  is the  $i$ -th species of DNA sequences.
- $m \in \{A, C, G, T\}^n$  is the input query sequence where  $n$  is the length of the input sequence.
- $\hat{m} \in \{A, C, G, T\}^\ell$  is the encrypted value of  $m$  where  $\ell$  is the length of the encrypted sequence.
- $E$  is an encryption function. Encryption function  $E$  takes input  $m$  and returns the encrypted sequence  $\hat{m} : E(m)$ . The encrypted sequence is valid if  $\hat{m} \in \{A, C, G, T\}^\ell$ .
- $\mathbf{M}_{D_i}$  is a trained model that takes target species  $D_i$  as training input.
- $y$  is an averaged probability output given by the trained model  $\mathbf{M}_{D_i}(m) \rightarrow p_m$  given input  $m$  where  $m \in D_i$ .
- $\hat{y}$  is a probability output given by the trained model  $\mathbf{M}_{D_i}(\hat{m}) \rightarrow p_{\hat{m}}$  presents given input  $\hat{m}$ , where  $\hat{m} \in \hat{D}_i$ .
- $\mathcal{A}$  is a probabilistic polynomial-time adversary. The adversary is an attacker that queries messages to the oracle model.
- $\epsilon$  is the standard deviation value of  $y$ .

## 3.2 Proposed DNA Steganalysis

A random oracle model [6] offers a middle ground between a full proof of security and no proof at all. Through the random oracle model, the encryption scheme  $E$  can be evaluated by querying an oracle that returns  $\hat{m}$  given input  $m$ . Assume that we have the random oracle that acts like current steganography scheme  $E$  such that any adversary  $\mathcal{A}$  breaks the random oracle with only negligible probability. By this assumption, steganography scheme  $E$  can be bounded by the soundness design [12].

The security of the random oracle is based on an *experiment* involving an adversary  $\mathcal{A}$ , and  $\mathcal{A}$ 's indistinguishability of the two encryptions. The experiment can be defined for any encryption scheme  $E$  over message space  $\mathbf{D}$  and for adversary  $\mathcal{A}$ . The *experiment* is defined as follows:

- (1) The random oracle chooses a random steganography scheme  $E$ . Scheme  $E$  modifies or extends the process of mapping a sequence of length  $n$  input to a sequence of length  $n$  of random sequence as the output. A process of mapping sequences can

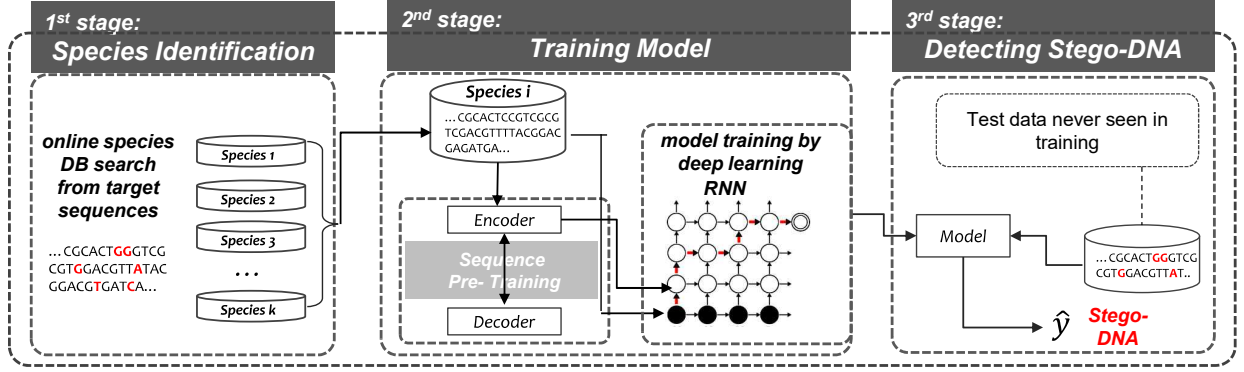


Figure 2: Overview of our proposed model (best viewed in color).

be viewed as a table indicating for each possible input  $m$  to the corresponding output value  $\hat{m}$ .

- (2) Adversary  $\mathcal{A}$  chooses a pair of sequences  $m_0, m_1 \in D_i$ .
- (3) The random oracle picks a bit  $b \in \{0, 1\}$  and sends encrypted message  $\hat{m} := E(m_b)$  to the adversary.
- (4) The adversary outputs a bit  $b'$ .
- (5) The output of the experiment is defined as 1 if  $b' = b$ , and 0 otherwise.  $\mathcal{A}$  succeeds in the *experiment* in the case of distinguishing  $m_b$ .

With the *experiment*, the definition of perfect security for  $E$  takes the following general form:

**Definition 3.1.** The scheme  $E$  is perfectly secure over message space  $\mathbf{D}$  if for every adversary  $\mathcal{A}$  it satisfies

$$\Pr[\text{experiment}] = \frac{1}{2}. \quad (1)$$

In the encryption scheme  $E$ ,  $\mathcal{A}$  cannot distinguish between  $m_0$  and  $m_1$ .  $\mathcal{A}$  learns no information about the presence of a hidden message.

In the real-world, most systems do not have access to a random oracle. Thus, pseudorandom function is typically applied by replacing the random function, which remains secure using soundness design. With the assumption, the oracle is replaced by a fixed encryption scheme  $E$  which corresponds to a transformation of a real system (implementation of the encryption scheme). An implementation of a random oracle is determined to be secure if the success probability of random oracle attack is negligible. Moreover, encryption scheme  $E$  is soundness secure if adversary  $\mathcal{A}$  has a *success* probability such that

$$\Pr[\text{success}] \leq \frac{1}{2} + \text{negligible}. \quad (2)$$

Using this notion of an implementation, we demonstrate breaking previous *experiment* using proposed method detecting hidden messages.

- (1) We construct  $\mathbf{M}_{D_i}$  that runs on random oracle where selected species  $D_i \in \mathbf{D}$ . Note that a model  $\mathbf{M}$  can be based on any classification models, but the key to select a model is to reduce the sparsity. Our proposed model  $\mathbf{M}$  is described in Section 3.4.
- (2) Adversary  $\mathcal{A}$  computes a standard deviation value of  $p_m$ .

- (3)  $\mathcal{A}$  computes  $y$  using  $\mathbf{M}_{D_i}(m_i)$  given  $m_i \in D_i$ .
- (4)  $\mathcal{A}$  computes  $\hat{y}$  using  $\mathbf{M}_{D_i}(\hat{m})$  given the output  $\hat{m}$ .
- (5) The  $\hat{m}$  is successfully detected if  $y - \hat{y} > \epsilon$ .

This gives a probability of two independent  $y$  and  $\hat{y}$  from  $\mathbf{M}_{D_i}$ . In section 4, we show that output messages are distinguishable through our proposed RNNs model with a success probability greater than  $\frac{1}{2} + \epsilon$ .

In the following, we evaluate the security of steganography scheme using information theoretical proof based on entropy  $H$  [8].

**LEMMA 3.2.** DNA steganography scheme is not secure if  $H(\mathbf{D}) > H(\hat{\mathbf{D}}|\mathbf{D})$ .

*Proof.* The mutual joint entropy  $H(\mathbf{D}, \hat{\mathbf{D}}) = H(\mathbf{D}) + H(\hat{\mathbf{D}}|\mathbf{D})$  is the union of both entropies for distribution  $\mathbf{D}$  and  $\hat{\mathbf{D}}$ . According to Gallager [17], the mutual information of  $I(\mathbf{D}; \hat{\mathbf{D}})$  is represented  $I(\mathbf{D}; \hat{\mathbf{D}}) = H(\mathbf{D}) - H(\mathbf{D}|\hat{\mathbf{D}})$ . It is symmetric in  $\mathbf{D}$  and  $\hat{\mathbf{D}}$  such that  $I(\mathbf{D}; \hat{\mathbf{D}}) = I(\hat{\mathbf{D}}; \mathbf{D})$ , and always non-negative. The conditional entropy between two distribution is 0 if and only if the distributions are equal. Thus, the mutual information must be zero to define secure DNA steganography schemes:

$$I(\mathbf{C}; (\mathbf{D}, \hat{\mathbf{D}})) = H(\mathbf{C}) - H(\mathbf{C}|\mathbf{D}, \hat{\mathbf{D}}) = 0. \quad (3)$$

where  $\mathbf{C}$  is message hiding space and it follows that:

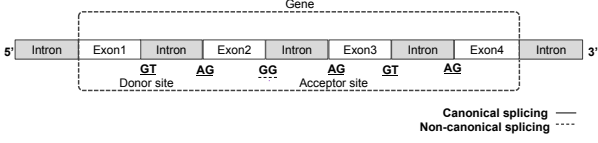
$$H(\mathbf{C}) = H(\mathbf{C}|\mathbf{D}, \hat{\mathbf{D}}). \quad (4)$$

Eq (3) means that the amount of entropy  $H(\mathbf{C})$  must not be decreased based on the knowledge of  $\mathbf{D}$  and  $\hat{\mathbf{D}}$ . It follows that the secure steganography scheme is obtained if and only if:

$$\forall i \in \mathbb{N}, m_i \in \mathbf{D}, \hat{m}_i \in \hat{\mathbf{D}} : m_i = \hat{m}_i.$$

Considering that representations of  $\hat{m}$  are limited, satisfying the condition is nearly impossible because the current steganography schemes are all based on the assumption of addition or substitution. Because  $\mathbf{C}$  is independent of  $\mathbf{D}$ , the amount of information will increase over distribution  $\hat{\mathbf{D}}$  if hidden messages are inserted over distribution  $\hat{\mathbf{D}}$ . We can conclude that the schemes are not secure under condition  $H(\mathbf{C}) > H(\mathbf{C}|\mathbf{D}, \hat{\mathbf{D}})$ .  $\square$

This means that one should not be surprised if a model exists that breaks the encryption scheme of a random oracle.



**Figure 3: Splicing signals in DNA sequence: Dimer GG represents non-canonical sites. Dimers GT and AG represent canonical donor and acceptor splice sites, respectively [31].**

### 3.3 Candidate Steganalysis Models

The proposed methodology can be applied to sequence analysis (i.e., denoising tools, sequence alignment, and splice junction classification [22, 30, 52]) in order to detect hidden messages. However, these methods often are restricted by the robustness of the target prediction and frequently miss important signals (GT or AT) during splitting and merging procedures [33].

However, intron and exon regions are utilized based on whether the task is one of data storage or transport as previously explained. We should note that splice junction prediction approach based on intron/exon modeling is more suitable for detecting hidden messages in both regions compared to the sequence analysis method. This because the splice junction prediction can detect characteristics of similar patterns in canonical/non-canonical splice sites.

Our previous work of splice junction Lee et al. [31] LSTM-based architecture considerably outperformed finding canonical/non canonical signals compared to existing alternatives regarding prediction accuracy with the UCSC database [27]. Hence, we adopt the LSTM-based architecture as the base model for DNA steganalysis. The model consists of three layers: input, hidden, and output. The input layer is connected to the hidden layer, which is composed of RNNs in order to model the internal structure of the DNA sequence. The outputs of the RNNs layer are fed into a fully connected output layer, which contains two units for classifying coding and non-coding regions. The details of detection of the steganography is explained in the following sections.

### 3.4 Proposed Steganalysis Model

Our approach is based on sequence-to-sequence learning using an autoencoder and stacked RNNs [42] which proceeds in two main steps: a) unsupervised pre-training of sequence-to-sequence autoencoder for modeling an overcomplete case, and b) supervised fine-tuning of stacked RNNs for modeling patterns between canonical and non-canonical splice sites. As shown in Figure 3, the splice junction site contains consensus strings called canonical splicing patterns and the most frequent patterns are dimer GT (called donor) and dimer AG at introns/exons boundaries [11]. The boundary between exons and introns referred to as the splice junction site.

In the proposed model, we use a set of DNA sequences that label introns and exons. These sequences are converted into a binary vector by orthogonal encoding [4]. It employs  $n_c$ -bit one-hot encoding. For  $n_c = 4$ ,  $\{A, C, T, G\}$  is encoded by

$$([1, 0, 0, 0], [0, 1, 0, 0], [0, 0, 1, 0], [0, 0, 0, 1]). \quad (5)$$

For example, the sequence ATTT is encoded into 16 dimensional binary vector  $\langle [1, 0, 0, 0], [0, 0, 0, 1], [0, 0, 0, 1], [0, 0, 0, 1] \rangle$ . The encoded sequence,  $\mathbf{x}$  where  $x$  is a tuple of  $n_c$  four-dimensional dense vector, is connected to the first layer of an autoencoder, which is used for unsupervised pre-training of sequence-to-sequence learning. An autoencoder is an artificial neural network used to learn meaningful encoding for a set of data in a case involving unsupervised learning. An autoencoder consists of two components; an encoder and decoder. The encoder RNN encodes  $\mathbf{x}$  to  $\mathbf{h}$ , and the decoder RNN decodes  $\mathbf{h}$  to the reconstructed  $\hat{\mathbf{x}}$  thus minimizing the reconstruction errors

$$\mathcal{L}(\mathbf{x}, \hat{\mathbf{x}}) = \|\mathbf{x} - \hat{\mathbf{x}}\|^2 \quad (6)$$

where  $\mathbf{h}$  is the representation of sequence features learned by features. Through unsupervised learning of the encoder-decoder model [49], we obtain representations of inherent features, which are directly connected to the second activation layer. The second layer is RNNs layer used to construct the model. The model in turn is used to determine patterns between canonical and non-canonical splice signals. We then obtain the tuple of

$$\mathbf{h} = \langle \mathbf{h}_1, \dots, \mathbf{h}_d \rangle \quad (7)$$

which is a representation of introns and exons in hidden layers, where  $d$  are a dimension of a vector. The features  $\mathbf{h}$  learned from the autoencoder are connected to the second stacked RNN layer, which consists of our proposed architecture for outputting classification probability for the given sequence  $D_i \in \mathbf{D}$ . For the fully connected output layer, we used the sigmoid function as the activation. The activation probability is given by

$$\Pr(Y = i|\mathbf{h}) = \frac{1/(1 + \exp(-\mathbf{w}_i^T \mathbf{h}))}{\sum_{k=0}^1 1/(1 + \exp(-\mathbf{w}_k^T \mathbf{h}))} \quad (8)$$

where  $Y$  is the label that indicates whether the given region contains introns ( $Y = 1$ ) or exons ( $Y = 0$ ). We used a recently proposed optimizer of multi-class logarithmic loss function (Adam [29]) for our training model. The objective function  $\mathcal{L}(\mathbf{w})$  that must be minimized is defined as follows:

$$\mathcal{L}(\mathbf{w}) = -\frac{1}{N} \sum_{n=1}^N (y_i \log(p_i) + (1 - y_i) \log(1 - p_i)) \quad (9)$$

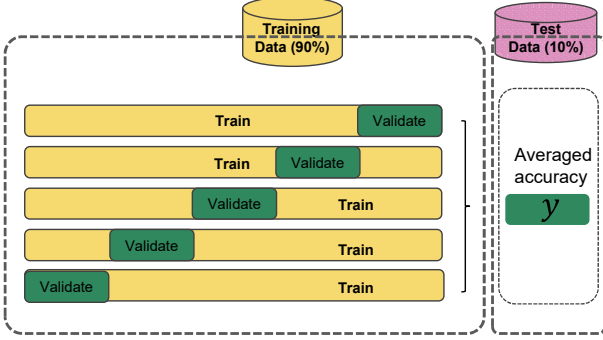
where  $N$  is the mini-batch size. A model  $\mathbf{M}_{D_i}$  has a possible probability of  $p_i$  for one species, where  $p_i$  is the probability of given non perturbed sequences being a valid intron-exon target sequences.

## 4 RESULTS

### 4.1 Experiment Setup

**4.1.1 Dataset.** We simulated our approach with the human UCSC-hg38 dataset [27], which includes sequences from 24 human chromosomes (22 autosomes and 2 sex chromosomes). The UCSC-hg38 dataset has a three-class classification (donor, acceptor, and non-site) and contains 24,279 genes with 1 to 173 (on average 9.44) exons per gene. We randomly selected 63,454 of 229,225 unique exons to remove duplicate exons caused by alternative splicing [28]. In addition, three samples were generated according to Noordewier





**Figure 4: 5-fold cross-validation:** For each machine learning approach to be trained, we randomly sampled 90% of the UCSC-hg38 data set for each fold of a five-way random split of the training data. The remaining 10% of the data set is used to compute averaged accuracy and standard deviation value across 5-fold cross-validation.

et al. [40] by taking the sequences from the center of exon’s left and right boundaries, representing acceptor, non-site, and donor.

**4.1.2 Input Representation.** The machine learning approach typically employs numerical representation of input for downstream processing. Orthogonal encoding, such as one-hot coding [4], is widely used to convert DNA sequences into numerical format. It employs  $n_c$ -bit one-hot encoding. For  $n_c = 4$ , {A, C, T, G} is encoded as described in Eq (5).

According to Lee et al. [31], the vanilla one-hot encoding scheme tends to limit generalization because of its sparsity of encoding (75% of the elements are zero). Thus, our approach encodes nucleotides into a four-dimensional dense vector that follows the direct architecture of a normal neural network layer [13], which is trained by the gradient decent method.

**4.1.3 Training.** The proposed RNN-based approach uses unsupervised training (4-30-50-4) for the autoencoder and supervised training (4-60-100-4) for the fine-tuning. In the unsupervised-training, the first number represent the number of input layers and the middle numbers indicate the number of units in the hidden RNNs layer and epochs. The last number represents the number of output units which is connected to the activations of the second layer.

For supervised-training, the first number represents the number of input layers, which is connected to stacked LSTM layers with full version including forget gates and peephole connections. The middle numbers indicate the numbers of units in the hidden RNNs layers and epochs. The last number represents the number of output layers, which is a fully connected output layer containing  $K$  units for  $K$ -class junction prediction. In our experiment, we used  $K = 3$  to classify sites (donors, acceptors and non-site). For the fully connected output layer, we used the sigmoid function for the activation. We used a recently proposed optimizer of multi-class logarithmic loss function Adam [29] for our training model. The objective function  $\mathcal{L}(\mathbf{w})$  that has to be minimized is as described

**Table 2: Comparison of detection performances for various sample sizes.** If hidden messages are detected in all regions for all modification rates, they are marked as o, otherwise, they are marked x.

	Both Region		Introns Region		Exons Region	
	Fully detected	Partially detected	Fully detected	Partially detected	Fully detected	Partially detected
RNN(proposed)	O	O	O	O	O	O
SVM [50]	X	O	X	O	O	O
Adaptive boosting [44]	X	O	X	O	X	O
Random forest[9]	O	O	X	O	O	O
Coral[43]	X	X	X	X	X	X
Lighter[48]	X	X	X	X	X	X
Sequence-alignment[16]	X	O	X	O	X	O

Eq( 9). We used a batch size of 100 and followed the most popularized batch normalization [25]. We initialized weights according to a uniform distribution as directed by Glorot and Bengio [20].

**4.1.4 Performance Evaluation.** We evaluated the performance of our proposed method based on four supervised learning algorithms (RNNs, SVM, random forests, and adaptive boosting) for detecting hidden messages. For the performance metric, we used accuracy differences in which accuracy<sup>1</sup> is the widely used measure. With the performance matrix, we evaluated learning algorithms with respect to the following three regions listed in Table 1; introns dedicated, exons dedicated, and both regions together.

For each algorithm, we generated simulated data for a different number of samples (100,200,300,400,500, and 1000) using the UCSC-hg38 dataset [27]. We also randomly selected 100 cases for the fixed sample size for each modification rate according to Table 1. With selected samples, we obtained the average prediction accuracy of a different number of samples against non-perturbed samples for 100 randomly selected cases. In the next step, we used the generated data according to the Table 1 modification rate to obtain prediction accuracy for 10 cases. Using averaged prediction accuracy of perturbed and non-perturbed, we checked the differences between the prediction accuracy rates for a different number of samples. As shown in Figure 4, we carried our 5-fold cross-validation to obtain mean/variance of accuracy differences.

## 4.2 Detection Performance on Sample Sizes

Figure 5 shows an experiment for each algorithm using six sample sizes (100, 200, 300, 400, 500 and 1000). Each algorithm was compared to three different regions based on the six sample sizes. The experiments were conducted by changing one percent of the hidden message. SVM showed good detection performance in the exon region but showed inferior performance in the intron and all regions. In the case of adaptive boosting, the detection performance was similar in both regions and in only intron regions, but performed poorly in exon regions. In the case of the random forest, exon and both regions showed good performance except for some modification rates. In intron regions, the detection performance

<sup>1</sup>Accuracy =  $(TP + TN)/(TP + TN + FP + FN)$ , where  $TP$ ,  $FP$ ,  $FN$ , and  $TN$  represent the numbers of true positives, false positives, false negatives, and true negatives, respectively.

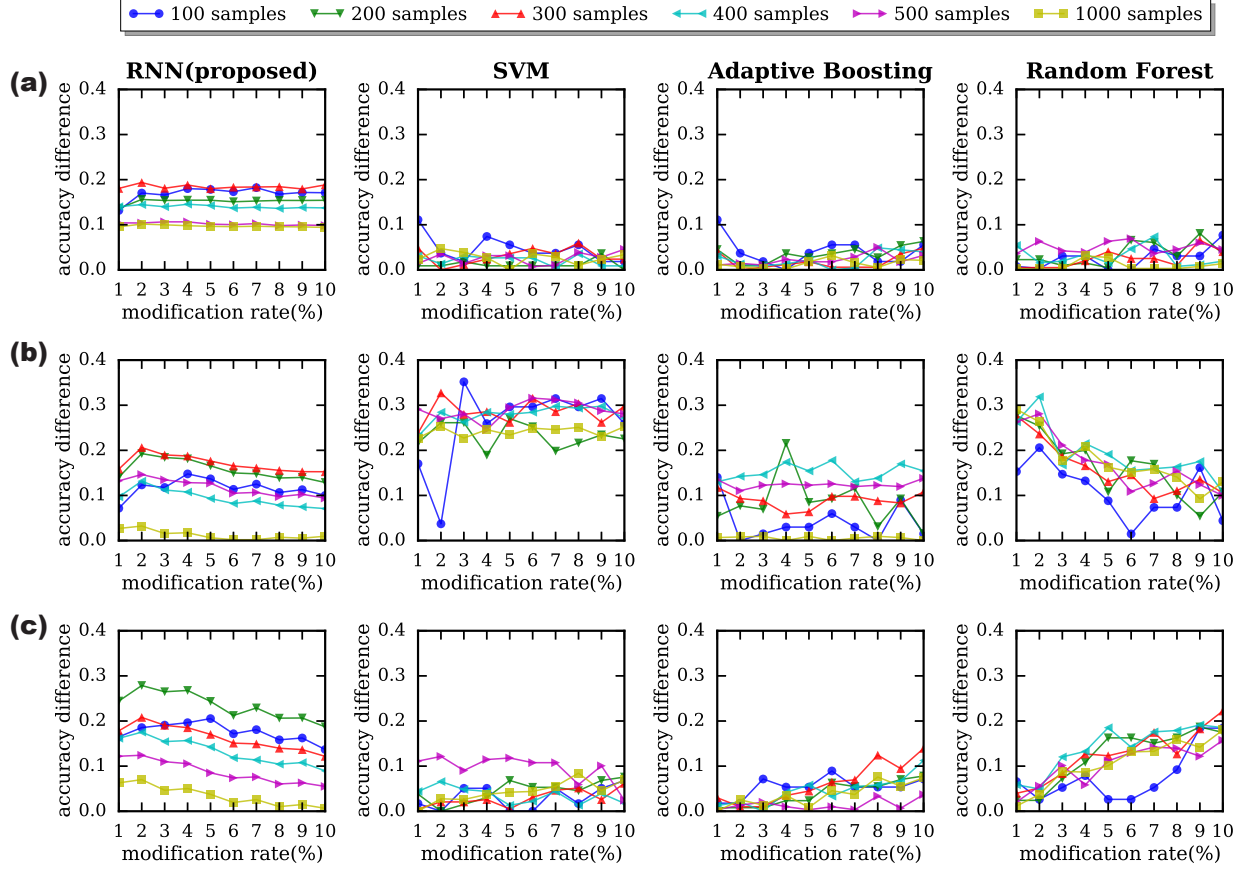


Figure 5: Comparison of learning algorithms with random hiding algorithms: (a) accuracy differences for introns region (b) accuracy difference for exons region (c) accuracy difference of both region. The performances of four supervised learning algorithms for detecting hidden messages (best viewed in color) are shown. Each algorithm is compared to three different regions for six sample sizes. In addition the experiments conducted by changing one percent of the hidden message.

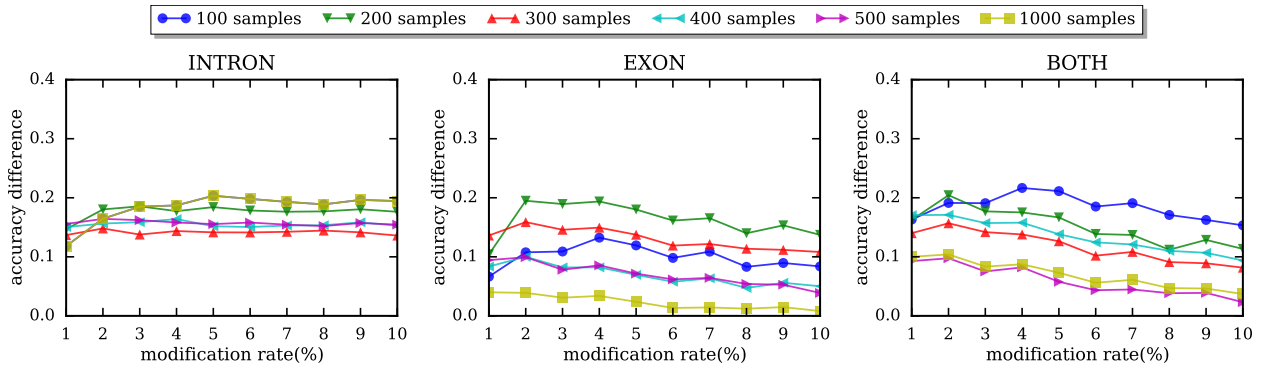


Figure 6: Detection performances using the complementary pair hiding algorithm for six sample sizes: 500 random DNA sequences are selected from the UCSC-hg38 dataset for different numbers of sample sizes in three regions. Each sample is perturbed by the complementary DNA sequences for modification rates from 1 to 10 (best viewed in color).

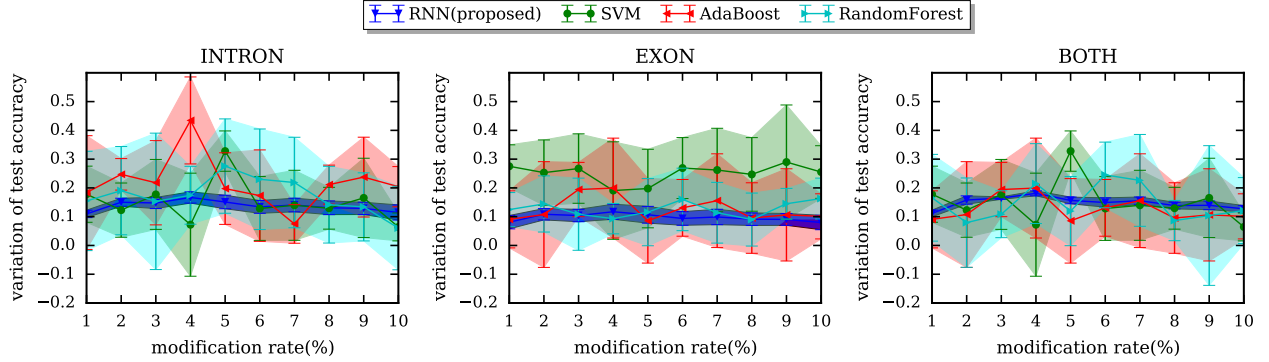


Figure 7: Comparison of learning algorithms in terms of robustness (best viewed in color). Mean and variance of accuracy are measured for the fixed sample size of 100 for 500 cases by changing one percent of the hidden message. Shared line represents standard deviation of inference accuracy.

Table 3: Detection performance of sequence alignment and denoising tools: 100 random DNA sequences are selected from the UCSC-hg38 dataset for sample sizes in three regions. One percent perturbed hidden messages are embedded by means of the complementary pair algorithm.

	Both Regions (%)	Introns Region (%)	Exons Region (%)
RNN(proposed)	99.93	99.96	99.94
Sequence-alignment[16]	84.00	85.00	85.00
Coral [43]	0.00	0.00	0.00
Lighter [48]	0.00	0.00	0.00

was similar to that of other learning algorithms. However, our proposed methodology based on RNNs outperformed hidden messages detection conducted in all regions.

Table 2 shows the detection performance of the learning algorithms with respect to three dedicated regions. As listed in Table 2, RNNs detected with all modification rates having a different number of samples, but other learning algorithms were incapable of full detection. Looking at individual metrics, SVM failed to detect with modification rates of 1,2,5, and 6% in introns and with 2 and 5% in exons. Adaptive boosting failed to detect with a modification rates of 1 and 3% in introns: 2,4, and 6% in exons: 2,4,6,8, and 10% in both regions. Random forest failed at detection with modification rates of 5 and 6% in exon regions. We carried 5-fold cross validation and the average performance was used to evaluate the learning classifier’s performance, as shown in Table 2.

In addition, we examined our proposed methodology based on denoising methods using coral and lighter. The UCSC-hg38 dataset was used to preserve local base structures and perturbed data samples were used as random noise. As shown in Table 3, results showed that both coral and lighter missed detection for all modification rates in all regions. Also, the method of sequence alignment performed poorly. The results suggest 15 to 16% chance exists that hidden messages may not be detected in all three regions.

We selected 500 random DNA sequences from the UCSC-hg38 dataset for different numbers of sample sizes in three regions. As described in Section 2.1, each sample was perturbed by the complementary DNA sequences [47] for modification rates from 1 to

10 by changing one percent of the message. The detection analysis using the complementary pair algorithm [47] complements our proposed algorithm with RNNs. Figure 6 shows the mean of accuracy differences between perturbed and non-perturbed sequences across 5-fold cross validation. The sampling was repeated 100 times, yielding 500 samples. The results shows that our proposed method, when compared with RNN, yields significantly better detection performances even with a complementary pair algorithm.

### 4.3 Variation of Detection Performance

To validate the learning algorithms with respect to robustness, we tested them with a fixed sample size of 100 with 500 cases for each modification rate to measure mean and variance of test accuracy. Figure 7 shows how the performance measures (mean and variance of accuracy differences) changes for modification rates from 1 to 10 in the introns, exons, and both regions. Each plotted entry is an averaged means over the 500 cases, and shade lines show an average of variances over 500 cases. The results suggest that hidden messages may not be detected if the prediction difference is less than variance. The overall analysis with respect to of robustness showed that the learning algorithms of SVM, random forests and adaptive boosting performed poorly. RNNs was the only model compared to other learning algorithms that never exhibited performing excellent detection across all three regions with all modification rates.

## 5 DISCUSSION

Next-generation sequencing has reduced the price of personal genomics [45], and the discovery of the CRISPR-Cas9 gene has made it possible to reconstruct DNA sequences [24]. The technology is yet to simulate over artificial DNA. However, human DNA sequences may become an object to which we can apply DNA watermarking. In this work, we proposed the first general DNA steganalysis method to detect hidden messages in DNA sequences. Our experiments using the real hg38 human genome implicitly consider that unknown relevant sequences are also detectable because of the characteristics of similar patterns in non-canonical splice sites. The number of donors with GT pairs and acceptors with AG pairs were found to be 86.32% and 84.63%, respectively.



Machine learning based steganalysis takes advantage of the following facts:

- No prior statistical models are required.
- A reasonably accurate detection rate can be achieved.
- Construction of universal steganalysis is possible.
- Averaging process eliminates the non-stationarity of data problem.

Our experimental results reveal that any supervised learning methods can construct a classifier given stego and stego-free data in the training phase. Upon convergence of the training phase, the final classifier can be obtained. Although many advantages exist using machine learning techniques for detecting hidden messages, there remain improvements

- Parameters tuning are dependent on the steganalyst. For example, training epochs, learning rate, and size of the training set.
- Type II errors, the fail to detect hidden messages while they exist, are not controllable by the steganalyst.
- Identifying portions of images/sequences in which a message is hidden as well as message extractions are extremely difficult.

There is no doubt that the limited factors are not easily solvable. However, we expect that counter-measures will be developed to foil our detection scheme. The future development of such techniques will automate hyperparameter tuning. According to Alvarez and Salzmänn [1], the numbers of layers and neurons of deep networks can be determined using an additional group sparsity regularizer to the objective function. The sizes of vectors of grouped parameters of each neuron in each layer are incur penalties if the loss converges. The influenced neurons are removed if the neurons are assigned value of zero.

## 6 CONCLUSION

In this paper, we proposed the first DNA-steganalysis based on splice junction framework using RNNs. Our contributions can be summarized as follows:

- A new DNA-steganalysis method is proposed using RNNs.
- Increased robustness is achieved using unsupervised to supervised learning.
- The improved method detects hidden messages while finding canonical/non canonical signals that are often missed by sequence analysis framework.

We believe that adopting our methodology will extend the field of DNA-steganalysis. For a future work, we plan to extend our model using automated hyperparameter tuning and applying it to diverse datasets in DNA sequences.

## ACKNOWLEDGMENTS

This work was supported in part by the National Research Foundation of Korea (NRF) grant funded by the Korea government (Ministry of Science, ICT and Future Planning) [No. 2014M3C9A3063541 and No. 2014M3A9E2064434], in part by a grant of the Korea Health Technology R&D Project through the Korea Health Industry Development Institute (KHIDI), funded by the Ministry of Health & Welfare [HI15C3224], and in part by the Brain Korea 21 Plus Project in 2017.

## REFERENCES

- [1] Jose M Alvarez and Mathieu Salzmann. 2016. Learning the Number of Neurons in Deep Networks. In *Advances in Neural Information Processing Systems*. 2262–2270.
- [2] Ross Anderson. 1996. *Information Hiding: First International Workshop, Cambridge, UK, May 30-June 1, 1996. Proceedings*. Vol. 1. Springer Science & Business Media.
- [3] Masanori Arita and Yoshiaki Ohashi. 2004. Secret signatures inside genomic DNA. *Biotechnology progress* 20, 5 (2004), 1605–1607.
- [4] Pierre Baldi and Søren Brunak. 2001. *Bioinformatics: the machine learning approach*. MIT press.
- [5] Marc B Beck, Eric C Rouchka, and Roman V Yampolskiy. 2012. Finding data in DNA: computer forensic investigations of living organisms. In *International Conference on Digital Forensics and Cyber Crime*. Springer, 204–219.
- [6] Mihir Bellare and Phillip Rogaway. 1993. Random oracles are practical: A paradigm for designing efficient protocols. In *Proceedings of the 1st ACM conference on Computer and communications security*. ACM, 62–73.
- [7] Krista Bennett. 2004. Linguistic steganography: Survey, analysis, and robustness concerns for hiding information in text. (2004).
- [8] Richard E Blahut. 1987. *Principles and practice of information theory*. Addison-Wesley Longman Publishing Co., Inc.
- [9] Leo Breiman. 2001. Random forests. *Machine learning* 45, 1 (2001), 5–32.
- [10] Sydney Brenner, Steven R Williams, Eric H Vermaas, Thorsten Störck, Keith Moon, Christie McCollum, Jen-I Mao, Shujun Luo, James J Kirchner, Sam Eletr, and others. 2000. In vitro cloning of complex mixtures of DNA on microbeads: physical separation of differentially expressed cDNAs. *Proceedings of the National Academy of Sciences* 97, 4 (2000), 1665–1670.
- [11] M Burset, IA Seledtsov, and VV Solovyyev. 2000. Analysis of canonical and non-canonical splice sites in mammalian genomes. *Nucleic acids research* 28, 21 (2000), 4364–4375.
- [12] Ran Canetti, Oded Goldreich, and Shai Halevi. 2004. The random oracle methodology, revisited. *Journal of the ACM (JACM)* 51, 4 (2004), 557–594.
- [13] François Chollet. 2015. Keras: Deep learning library for theano and tensorflow. (2015).
- [14] Catherine Taylor Clelland, Viviana Risca, and Carter Bancroft. 1999. Hiding messages in DNA microdots. *Nature* 399, 6736 (1999), 533–534.
- [15] Heather J Cordell and David G Clayton. 2005. Genetic association studies. *The Lancet* 366, 9491 (2005), 1121–1131.
- [16] Robert C Edgar. 2010. Search and clustering orders of magnitude faster than BLAST. *Bioinformatics* 26, 19 (2010), 2460–2461.
- [17] Robert G Gallager. 1968. *Information theory and reliable communication*. Vol. 2. Springer.
- [18] Ashish Gehani, Thomas LaBean, and John Reif. 2003. DNA-based cryptography. In *Aspects of Molecular Computing*. Springer, 167–188.
- [19] Daniel G Gibson, John I Glass, Carole Lartigue, Vladimir N Noskov, Ray-Yuan Chuang, Mikkel A Algire, Gwynedd A Benders, Michael G Montague, Li Ma, Monzia M Moodie, and others. 2010. Creation of a bacterial cell controlled by a chemically synthesized genome. *science* 329, 5987 (2010), 52–56.
- [20] Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *Aistats*, Vol. 9. 249–256.
- [21] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. *Deep Learning*. MIT Press. <http://www.deeplearningbook.org>.
- [22] Karol Gregor, Ivo Danihelka, Alex Graves, and Daan Wierstra. 2015. DRAW: A recurrent neural network for image generation. *arXiv preprint arXiv:1502.04623* (2015).
- [23] Dominik Heider and Angelika Barnekow. 2007. DNA-based watermarks using the DNA-Crypt algorithm. *BMC bioinformatics* 8, 1 (2007), 1.
- [24] Patrick D Hsu, Eric S Lander, and Feng Zhang. 2014. Development and applications of CRISPR-Cas9 for genome engineering. *Cell* 157, 6 (2014), 1262–1278.
- [25] Sergey Ioffe and Christian Szegedy. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167* (2015).
- [26] Shuhong Jiao and Robert Goutte. 2008. Code for encryption hiding data into genomic DNA of living organisms. In *2008 9th International Conference on Signal Processing*. IEEE, 2166–2169.
- [27] W James Kent, Charles W Sugnet, Terrence S Furey, Krishna M Roskin, Tom H Pringle, Alan M Zahler, and David Haussler. 2002. The human genome browser at UCSC. *Genome research* 12, 6 (2002), 996–1006.
- [28] Hadas Keren, Galit Lev-Maor, and Gil Ast. 2010. Alternative splicing and evolution: diversification, exon definition and function. *Nature Reviews Genetics* 11, 5 (2010), 345–355.
- [29] Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).
- [30] Diederik P Kingma and Max Welling. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114* (2013).

- [31] Byunghan Lee, Taehoon Lee, Byunggook Na, and Sungroh Yoon. 2015. DNA-Level Splice Junction Prediction using Deep Recurrent Neural Networks. *arXiv preprint arXiv:1512.05135* (2015).
- [32] Byunghan Lee, Taesup Moon, Sungroh Yoon, and Tsachy Weissman. 2015. DUDE-Seq: Fast, flexible, and robust denoising of nucleotide sequences. *arXiv preprint arXiv:1511.04836* (2015).
- [33] Taehoon Lee and Sungroh Yoon. 2015. Boosted Categorical Restricted Boltzmann Machine for Computational Prediction of Splice Junctions. In *ICML*. 2483–2492.
- [34] Taehoon Lee and Sungroh Yoon. 2015. Boosted Categorical Restricted Boltzmann Machine for Computational Prediction of Splice Junctions. In *ICML*. 2483–2492.
- [35] André Leier, Christoph Richter, Wolfgang Banzhaf, and Hilmar Rauhe. 2000. Cryptography with DNA binary strands. *Biosystems* 57, 1 (2000), 13–22.
- [36] David J Lockhart and Elizabeth A Winzeler. 2000. Genomics, gene expression and DNA arrays. *Nature* 405, 6788 (2000), 827–836.
- [37] Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of Machine Learning Research* 9, Nov (2008), 2579–2605.
- [38] Indra Kanta Maitra. 2011. Digital steganalysis: Review on recent approaches. *Journal of Global Research in Computer Science* 2, 1 (2011).
- [39] Seonwoo Min, Byunghan Lee, and Sungroh Yoon. 2016. Deep learning in bioinformatics. *Briefings in Bioinformatics* (2016), bbw068.
- [40] Michiel O Noordewier, Geoffrey G Towell, and Jude W Shavlik. 1991. Training knowledge-based neural networks to recognize genes in DNA sequences. *Advances in neural information processing systems* 3 (1991), 530–536.
- [41] Seunghyun Park, Seonwoo Min, Hyunsoo Choi, and Sungroh Yoon. 2016. deep-MiRGene: deep neural network based precursor microRNA prediction. *arXiv preprint arXiv:1605.00017* (2016).
- [42] Sarah M Peterson, Jeffrey A Thompson, Melanie L Ufkin, Pradeep Sathyanarayana, Lucy Liaw, and Clare Bates Congdon. 2014. Common features of microRNA target prediction tools. *Frontiers in genetics* 5 (2014), 23.
- [43] Leena Salmela. 2010. Correction of sequencing errors in a mixed set of reads. *Bioinformatics* 26, 10 (2010), 1284–1290.
- [44] Robert E Schapire, Yoav Freund, Peter Bartlett, Wee Sun Lee, and others. 1998. Boosting the margin: A new explanation for the effectiveness of voting methods. *The annals of statistics* 26, 5 (1998), 1651–1686.
- [45] Stephan C Schuster. 2008. Next-generation sequencing transforms today’s biology. *Nature methods* 5, 1 (2008), 16.
- [46] Boris Shimanovsky, Jessica Feng, and Miodrag Potkonjak. 2002. Hiding data in DNA. In *International Workshop on Information Hiding*. Springer, 373–386.
- [47] HJ Shiu, Ka-Lok Ng, Jywe-Fei Fang, Richard CT Lee, and Chien-Hung Huang. 2010. Data hiding methods based upon DNA sequences. *Information Sciences* 180, 11 (2010), 2196–2208.
- [48] Li Song, Liliana Florea, and Ben Langmead. 2014. Lighter: fast and memory-efficient sequencing error correction without counting. *Genome biology* 15, 11 (2014), 509.
- [49] Nitish Srivastava, Elman Mansimov, and Ruslan Salakhutdinov. 2015. Unsupervised Learning of Video Representations using LSTMs. In *ICML*. 843–852.
- [50] Johan AK Suykens and Joos Vandewalle. 1999. Least squares support vector machine classifiers. *Neural processing letters* 9, 3 (1999), 293–300.
- [51] Kazuo Tanaka, Akimitsu Okamoto, and Isao Saito. 2005. Public-key system using DNA as a one-way function for key distribution. *Biosystems* 81, 1 (2005), 25–29.
- [52] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. 2008. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*. ACM, 1096–1103.
- [53] Qinglong Wang, Wenbo Guo, Il Ororbia, G Alexander, Xinyu Xing, Lin Lin, C Lee Giles, Xue Liu, Peng Liu, and Gang Xiong. 2016. Using Non-invertible Data Transformations to Build Adversary-Resistant Deep Neural Networks. *arXiv preprint arXiv:1610.01934* (2016).
- [54] Pak Chung Wong, Kwong-kwok Wong, and Harlan Foote. 2003. Organic data memory using the DNA approach. *Commun. ACM* 46, 1 (2003), 95–98.
- [55] Nozomu Yachie, Kazuhide Sekiyama, Junichi Sugahara, Yoshiaki Ohashi, and Masaru Tomita. 2007. Alignment-Based Approach for Durable Data Storage into Living Organisms. *Biotechnology progress* 23, 2 (2007), 501–505.
- [56] Shu Yan, Fan Chen, and Hongjie He. 2016. Improved Separable Reversible Data Hiding in Encrypted Image Based on Neighborhood Prediction. In *International Conference on Cloud Computing and Security*. Springer, 94–103.
- [57] Xiao Yang, Sriram P Chockalingam, and Srinivas Aluru. 2013. A survey of error-correction methods for next-generation sequencing. *Briefings in bioinformatics* 14, 1 (2013), 56–66.
- [58] Xinpeng Zhang, Jing Long, Zichi Wang, and Hang Cheng. 2016. Lossless and reversible data hiding in encrypted images with public-key cryptography. *IEEE Transactions on Circuits and Systems for Video Technology* 26, 9 (2016), 1622–1631.